Office Action Summary	Application No. 09/512,592	Applicant(s) Bruce M. Dickens
	Examiner Jean R. Homere	Art Unit 2177

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136 (a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) [X] Responsive to communication(s) filed on *Feb 19, 2002*.

2a) [ ] This action is **FINAL**. 2b) [X] This action is non-final.

3) [ ] Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11; 453 O.G. 213.

Disposition of Claims

4) [X] Claim(s) *1-76* is/are pending in the application.

4a) Of the above, claim(s) ______ is/are withdrawn from consideration.

5) [ ] Claim(s) ______ is/are allowed.

6) [X] Claim(s) *1-76* is/are rejected.

7) [ ] Claim(s) ______ is/are objected to.

8) [ ] Claims ______ are subject to restriction and/or election requirement.

Application Papers

9) [ ] The specification is objected to by the Examiner.

10) [ ] The drawing(s) filed on ______ is/are a) [ ] accepted or b) [ ] objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11) [ ] The proposed drawing correction filed on ______ is: a) [ ] approved b) [ ] disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12) [X] The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13) [ ] Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) [ ] All b) [ ] Some* c) [ ] None of:

1. [ ] Certified copies of the priority documents have been received.
2. [ ] Certified copies of the priority documents have been received in Application No. ______.
3. [ ] Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

*See the attached detailed Office action for a list of the certified copies not received.

14) [ ] Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

a) [ ] The translation of the foreign language provisional application has been received.

15) [ ] Acknowledgement is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) [ ] Notice of References Cited (PTO-892)
2) [ ] Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) [ ] Information Disclosure Statement(s) (PTO-1449) Paper No(s). ______
4) [ ] Interview Summary (PTO-413) Paper No(s). ______
5) [ ] Notice of Informal Patent Application (PTO-152)
6) [ ] Other:

DETAILED ACTION

Response to Amendments

1. Applicant's arguments with respect to claims 1-76 have been considered but are moot in view of the new ground(s) of rejection. Applicant's comments suggested substantial differences from the original specification and pointed to claim language that, in light of the comments, raise several issues of written description and enablement that result in the newly introduced rejections under 35 USC 112 below.

Reissue Applications

Objection to Oath/Declaration

2a. The reissue oath/declaration filed with this application is defective because it fails to identify at least one error which is relied upon to support the reissue application. Further, it fails to refer to the amendment of 1/05/2002. See 37 CFR 1.175(a)(1), 37 CFR 1.63(b)(2), and MPEP 1414.

2b. The new Oath/Declaration submitted on 02/19/2002 is unsigned as required by MPEP 1401.01.

3. The original patent, or an affidavit or declaration as to loss or inaccessibility of the original patent, must be received before this reissue application can be allowed. See 37 CFR 1.178.

4. Changes made in the certificate of correction have not been incorporated into the specification of the reissue application. Applicant is required to submit a substitute specification which complies with reissue practice.

5a. This application is objected to under 37 CFR 1.172(a) as lacking the written consent of all assignees owning an undivided interest in the patent. The consent of the assignee must be in compliance with 37 CFR 1.172. See MPEP 1410.01. A proper assent of the assignee in compliance with 37 CFR 1.172 and 3.73 is required in reply to this Office action.

5b. This application is objected to under 37 CFR 1.172(a) as the assignee has not established its ownership interest in the patent for which reissue is being requested. An assignee must establish its ownership interest *in order to support the consent to a reissue application required by 37 CFR 1.172(a)*. The submission establishing the ownership interest of the assignee is informal. There is no indication of record that the party who signed the submission is an appropriate party to sign on behalf of the assignee. 37 CFR 3.73(b). A proper submission establishing ownership interest in the patent, pursuant to 37 CFR 1.172(a), is required in response to this action.

5c. The person who signed the submission establishing ownership interest has failed to state his/her capacity to sign for the corporation or other business entity, and he/she has

not been established as being authorized to act on behalf of the assignee. See MPEP § 324.

5d. It would be acceptable for a person, other than a recognized officer, to execute a submission establishing ownership interest, <u>provided</u> the record for the application includes a statement that the person is empowered to sign a submission establishing ownership interest and/or act on behalf of the organization.

5e. Accordingly, a new submission establishing ownership interest which includes such a statement above, will be considered to be executed by an appropriate official of the assignee. A separately filed paper referencing the previously filed submission establishing ownership interest and containing a proper empowerment statement would also be acceptable.

6. Claims 1-76 are rejected as being based upon a defective reissue declaration under 35 U.S.C. 251 as set forth above. See 37 CFR 1.175. The nature of the defect(s) in the declaration is set forth in the discussion above in this Office action.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the first and second paragraphs of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to

make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7i. Claims 16-67, 69-73, 75- 77 are rejected under 35 U.S.C. 112, first paragraph, as containing subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

7ii. Claims 1-76 are rejected under 35 U.S.C. 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter which the applicant regards as his invention.

A. Claims 32 and 69 call for sorting dates "in the form of C1C2Y1Y2", while the specification only describes sorting with the format C1C2Y1Y2M1M2D1D2. (col.2, lines 15-21, col.3, lines 38-48). These two sorting formats are different from each other since the former excludes month and date data from the sort keys, resulting in a faster sort, with a potentially different resultant sequence, than the latter, in which, unlike the latter, the runs of data for any given century and year combination are not further sorted by month and day.

One of ordinary skill in the art at the time of the invention would have immediately appreciated that a CCYYMMDD sort is fundamentally different than, as contrasted with being a species for the genus of, a CCYY sort because of these two dramatic differences in the sort results. Therefore, because a CCYY sort is not merely a broader genus for the species of CCYYMMDD, the disclosed CCYYMMDD sort does not implicitly satisfy the written description and enablement requirements under 35 USC 112, first paragraph. Therefore, the claimed sorting format is new matter since it is not disclosed in the original specification.

B. Claims 33, 60-61, 64-65 and 70 call for reformatting to occur "without changing " or "without modifying" the symbolic date representations during the reformatting when the specification merely indicates that the YYMMDD date format is reformatted to appear in the form CCYYMMDD (col.3, lines 41-43). It is apparent that the original specification is devoid of any disclosure of how such reformatting is performed "without changing" or "without modifying" the symbolic date representation. In fact, the suggestion of reformatting without changing representation is on its face a contradiction, for the reformat is to change representation. Therefore, the claimed limitation "reformatting to occur "without changing " or "without modifying"" is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

C. Claims 16-30, 32, 34-67, 69-71, 75 and 76 call for processing relative to a "pivot date" or "pivot year" when such terms are nowhere defined or even mentioned in the original

specification. Therefore, the claimed limitation "pivot date" or "pivot year" is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

D. Claims 20-21, 62-65 and 71 call for "reformatting" or "storing" "separately" from the symbolic representations in the database or from the database when the original specification merely suggests reformatting or sorting the date. However, the original specification does not disclose such "separate" reformatting or storing. Therefore, the claimed limitation of "separate storing " or "separate reformatting" is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

E. Claims 16-25, 31-33, 66-67 and 72 call for "collectively further processing" when the specification makes no mention of such "collective" further processing. Similarly, claims 36-43 call for "collectively sorting" or "collectively manipulating" when the original specification merely suggests sorting and manipulating. However, it does not mention such "collectively" sorting or manipulating. Similarly, claims 34-61, 63 and 65 call for the step of "running a program collectively" when the original specification, perhaps, only implicitly discloses the "running of the program". However, such "collective" running of the program, is not disclosed. Therefore, the claimed limitations of "collective processing", "collective sorting", "collective manipulating" or "collective running" are new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

F. Claims 36-37, 40-41, 48-49, 51-59, and 69 call for the running of a program after a sorting operation has been performed. However, the original specification does not provide a written description of such running of a program subsequent the step of sorting. Similarly, claims 38-39, 42-43 call for data manipulation before running of the program. No written description is provided for such data manipulation before running the program in the original specification. Therefore, such limitations are new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

G. Claims 46-59 call for "repeating the step of converting at least a substantial portion" of the specified data. The original specification does not disclose the conversion of such substantial portion. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

H. Claims 34-65 and 70-71 call for "converting " symbolic representations "by windowing the symbolic representation" when the specification merely discloses the selection of a 10 decade window. The verb "windowing" appears nowhere in the specification, and its meaning is unclear. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

I. Claims 35, 37, 39, 41, 43, 45, 49, 51, 53, 55, 57 and 59 call for the step of "opening the database prior to the step of converting" when the original specification makes no mention of

opening the database. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

J. Claims 34-65, 70 and 71 call for the avoidance of an "ambiguity" by reformatting or converting date representation. The original specification merely suggests that dates containing only two digit year representation, and without reformatting, may sort improperly. It does not mention or discuss any such claimed ambiguity. Therefore, such limitation is new matter because this subject matter was given neither a written description nor enabling description in the original disclosure.

K. Claims 1-15, 31, 33, 68, 72-74 call for the selection of a "YAYB value for the first decade" of a window. There is no known meaning for the "value of a decade" and the original specification is devoid of any description of what the "value of a decade" is. Because this subject matter was in the original disclosure, such limitation is not new matter. However, it is rejected under the second paragraph of 35 USC 112 because the meaning of the claim phraseology is so devoid as to be wholly indefinite.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter

sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-3, 5, 7, 9-10 are rejected under 35 U.S.C. 103(a) as obvious over Daniel P. Shaughnessy, US. Patent No. 5, 630,118, filed on November 21, 1994 and issued on May 13, 1997 (Shaughnessy, hereinafter) in view of Masakazu Hazama, Japanese Application No. 05-027947, published on February 5, 1993 (Hazama, hereinafter).

Because there are so many claims with so many subject matter elements, the detailed, claim by claim analysis may be too repetitive. In an effort to provide an overriding clarity to the following rejections, the following are noted:

- The Shaughnessy reference is an essentially complete teaching of the claimed subject matter. In particular, it teaches modifying those dates that have a two digit identifier less than some predetermined pivot date , changing the format of the date, and sorting the results. However, Shaughnessy does not explicitly state that the predetermined pivot date is less than any date in the database.
- Hazama is provided, consequently, as an explicit teaching of the need for the pivot date to be less than any date in the database.
- Therefore, it follows that, logically, the suggested process of converting all dates in

the database, wherein two digit dates are converted into four digit dates as taught in Shaughnessy cannot operate correctly unless the pivot date is less than any date in the database. This is due to the fact that any dates in the database that were less than the pivot date would be incorrectly altered to a date in the succeeding century.

- It also follows that this assignment of a pivot date is simply a species of the genus of setting program parameters according the specific input data criteria.
- Further, it follows that one of ordinary skill in the art of programming would know and would be adept at setting parameters to correctly process a set of data. Applicant is reminded that the software development process consists both of design, in which process is matched to the scope of the input, and testing, in which data are entered through the process to check results. Thus, either of which would have provided sufficient notice to the ordinary skilled artisan that setting the pivot date to accommodate the input data is a necessity.

- Although the following claims may be rejected over Shaughnessy alone, given the logical necessity of setting the pivot date properly, and having articulated and placed the above facts and analysis in the record consistent with the recent decision of *In re Lee*, 61 USPQ2d 1430 (CA FC 2002), the Hazama reference is provided to demonstrate that, apart from being logically necessary, this attribute of the pivot date being earlier than the dates in the database was, in fact, known to those of ordinary skill in the art at the time of the invention, and not a ground breaking discovery by the applicant.

As to claim 1, Shaughnessy substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq).

The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10_decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical

matter, incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[1] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2 , and D.sub.1 D.sub.2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq) and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection

---

[1] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation whereby a '10_decade window includes the decade beginning in the year 2000' by suggesting the use of a 100 year window that includes a decade date in the 21st century (col. 6, lines 28-29 et seq).

As to claim 3, Shaughnessy discloses the invention as discussed in the rejection of claim 2, as well as the claimed limitation whereby, 'the step of determining includes the step of

determining the first value as 20 and the second value as 19' by assigning the century value to 19 if the YYDDD portion of the date is greater than or equal to the corresponding portion of the corresponding portion of the modified system install date (col. 5, lines 40-46) and by assigning the century value to 20 if the pivot date is less than the modified system install date (col.5, lines 52-60 et seq).

As to claim 5, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein 'the step of reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2' as the conversion of the current date from a six digit format (YYMMDD) into an 8-digit format (CCYYMMDD) (col. 5, lines 48-50 et seq).

As to claim 7, Shaughnessy discloses the invention as discussed in the rejection of claim 1, as well as the claimed limitation, wherein the step of providing a database includes the step of converting pre_existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by as the converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 9, Shaughnessy discloses the invention as discussed in the rejection of claim 1,

as well as the claimed limitation of 'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting' by saving the converted date in the database (col. 6, lines 1-3 et seq). Also Shaughnessy col. 1 lines 31-35 and col. 4 lines 12-23 indicate this is in fact one solution to the Y2K problem, but suggests that it is an expensive solution - Shaughnessy teaches away only from an economic, not a technical viewpoint. The examiner takes administrative notice that this solution of storing the symbolic representations back in the database, again, taught by Shaughnessy as an available solution, is the only permanent solution, and is therefore inevitable - the economic rationale in Shaughnessy is temporary - eventually the data in a database spans over 100 years.

As to claim 10, Shaughnessy discloses the invention as discussed in the rejection of claim 9, as well as the claimed limitation of ' manipulating information in the database having the reformatted date information therein' by performing updates on the converted dates and saving said converted dates in the database (col. 6, lines 1-22 et seq).

10. Claims 4, 6, 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, as applied to the rejection of claims 1-3, 5, 7, 9-10 above, further in view of Booth et al., Implementation in Clipper 5A Developer's Guide (Booth, hereinafter).

As to claim 4, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 1 above. Shaughnessy and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy- Hazama's system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 6, Shaughnessy substantially discloses the invention as discussed in the rejection of claim 5 above. Shaughnessy and Hazama do not specifically disclose the additional step of 'sorting the symbolic representations of dates using a numerical order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes

the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy - Hazama's system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 8, Shaughnessy and Hazama substantially discloses the invention as discussed in the rejection of claim 1. Shaughnessy and Hazama do not specifically disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom. See p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting

a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2 M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Shaughnessy- Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

11. Claims 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 11, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq).

The method as recited in the body of the claim is particularly taught by the cited references as follows:

. Shaughnessy discloses the claimed step of 'providing a database with dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator, all of dates falling within a 10-decade period of time which includes the decade beginning in the year 2000' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date, and wherein the ten decade includes a decade date in the 21st century (col. 6, lines 28-29 et seq). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[2] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36).

Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby

---

[2]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, is therefore not any later than the earliest date in 100 year-cycle in the

determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[3] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting each date in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the

---

database.

[3] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the step of 'sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy-Hazama' s system to return the reformatted dates in chronological sequence. . Such

disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 12, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Additionally, Shaughnessy discloses step of ' converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 13, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Shaughnessy by disclosing the claimed step of 'selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Shaughnessy- Hazama' s system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 14, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Shaughnessy by disclosing the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would allow users of Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 15, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 14. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

12. Claims 16-18, 20, 22, 24-25 are rejected under 35 U.S.C. 103(a) as obvious over Shaughnessy in view Hazama.

As to claim 16, Shaughnessy substantially discloses the invention including the claimed

'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq).

The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and

industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[4] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[5] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic

---

[4]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[5] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 17, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 16. Further, Shaughnessy discloses the claimed limitation whereby 'the window includes at least a portion of the decade beginning in the year 2000' by suggesting the use of a 100 year window that includes a decade date in the 21st century (col. 6, lines 28-29 et seq).

As to claim 18, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 17. Further, Shaughnessy discloses the claimed limitation whereby, 'the step of determining includes the step of determining the first value as 20 and the second value as 19' by assigning the century value to 19 if the YYDDD portion of the date is greater than or equal to the corresponding portion of the corresponding portion of the modified system install date (col. 5, lines 40-46) and by assigning the century value to 20 if the pivot date is less than the modified system install date (col.5, lines 52-60 et seq).

As to claim 20, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 16,. Further, Shaughnessy discloses the claimed limitation, wherein 'the step of reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 separately from the symbolic representations in the database' as the conversion of the current date from a six digit format (YYMMDD) into an 8-digit format (CCYYMMDD) (col. 5, lines 48-50 et seq).

As to claim 22, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 16. Further, Shaughnessy discloses the claimed limitation, wherein the step of providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by as the converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 24, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 16,. Further, Shaughnessy discloses the claimed limitation of 'storing the symbolic representation of dates and their associated information back into the database after the step of reformatting' by saving the converted date in the database (col. 6, lines 1-3 et seq).

As to claim 25, Shaughnessy and Hazama disclose the invention as discussed in the rejection of claim 24. Further, Shaughnessy discloses the claimed limitation of ' manipulating information in the database having the reformatted date information therein' by performing updates on the converted dates and saving said converted dates in the database (col. 6, lines 1-22 et seq).

13. Claims 19, 21, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

As to claim 19, Shaughnessy and Hazama substantially disclose the invention as discussed in the rejection of claim 16 above. Shaughnessy and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 21, Shaughnessy and Hazama substantially disclose the invention as discussed in the rejection of claim 20 above. Shaughnessy and Hazama do not specifically disclose the additional step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama' s system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 23, Shaughnessy and Hazama substantially disclose the invention as discussed in the rejection of claim 16. Shaughnessy and Hazama do not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to

process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

14. Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 26, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a

two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq).

The method as recited in the body of the claim is particularly taught by the cited references as follows:

. Shaughnessy discloses the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Also, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date. Further, Shaughnessy discloses the claimed 'all of the symbolic representations of dates falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy also discloses a subroutine for determining the current date[6] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100

---

[6] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[7] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of ' reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines

than the earliest date in 100 year-cycle in the database.

[7] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the step of 'sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In

particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 27, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Additionally, Shaughnessy discloses step of 'converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by converting the current date in a six digit format (YYMMDD), wherein YY represents the year, MM represents the month and DD represents the day (col. 8, lines 18-27 et seq).

As to claim 28, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Shaughnessy and Hazama by disclosing the claimed step of 'selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of Shaughnessy-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 29, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Shaughnessy by disclosing the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 30, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 29. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to

one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

15. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 31, Shaughnessy substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq).

The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date having a

cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[8] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[9] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field

---

[8] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[9] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

. Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller

than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

16. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 32, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y.sub.1 Y.sub.2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines

11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[10] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[11] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each of the dates in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to

---

[10]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[11] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates

already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically, disclose the step of ' sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. . Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

17. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 33, Shaughnessy substantially discloses the invention including the claimed

'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[12] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century

[12]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than A YB and having a second value if Y1 Y2 is equal to or greater than A YB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[13] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database without changing any of the symbolic representations of a date in the database during the reformatting step, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without changing any of the symbolic representations of a date in the database (col.5, lines 46-51; col.6, lines 57-65 et seq), and by returning one date field with the converted date to the subroutine and a means for returning a parameter to the application program for use in further operations (col. 1, lines 47-54 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of

---

than the earliest date in 100 year-cycle in the database.

[13] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

18. Claims 34-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

method

As to claim 34, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database

utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date

falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[14] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[15] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database

---

[14]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[15] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET

EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

As to claim 35, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Shaughnessy discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit format (col. 4, lines 29-33 et seq).

As to claim 36, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 37, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of

'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 38, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 39, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 40, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field

contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 41, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 42, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in

chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 43, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data entry field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 44, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 45, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Shaughnessy by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 46, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Shaughnessy further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (col. 5, lines 48-50 et seq).

As to claim 47, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Shaughnessy further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date

field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (col. 5, lines 48-50 et seq).

As to claim 48, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 49, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Shaughnessy by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 50, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 51, Shaughnessy, Hazma and Booth disclose the invention as discussed in the rejection of claim 49. Booth further complements Shaughnessy and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 52, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Shaughnessy and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq),

whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 53, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Shaughnessy and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 54, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Shaughnessy and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 55, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Shaughnessy and Hazama by

disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 56, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Shaughnessy and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 57, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Shaughnessy and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq),

whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 58, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 54. Booth further complements Shaughnessy and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 59, Shaughnessy, Hazama and Booth disclose the invention as discussed in the rejection of claim 55. Booth further complements Shaughnessy and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

19. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 60, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such

windowing and converting; and running a program on each of the converted symbolic representations of each of the respective dates to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the date data symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[16] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[17] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD

[16]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[17] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et

seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

20. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 61, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without

mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the

current date[18] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[19] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if

[18] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[19] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy's

system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

21. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 62, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries'as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

· Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates

as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of-dates--contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[20] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[21] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending

[20]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[21] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However,

Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

Shaughnessy and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements the cited references by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

22. Claim 63 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 63, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a

program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[22] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[23] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD

[22]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[23] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et

seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

Shaughnessy and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements the cited references by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

23. Claim 64 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 64, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the

converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[24] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[25] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

[24]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[25] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to

determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

Shaughnessy and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements the cited references by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

24. Claim 65 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 65, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of ' converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing

a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[26] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[27] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of

[26] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[27] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date

(C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

Shaughnessy and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements the cited references by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

25. Claim 66 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 66, Shaughnessy substantially discloses the invention including the claimed 'Shaughnessy discloses the claimed 'method of processing dates in a database' as a method

and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[28] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy

[28]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[29] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, M1M2, and D1D2; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of

---

than the earliest date in 100 year-cycle in the database.

[29] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

26. Claim 67 is rejected under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 67, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database'

as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the

year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[30] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1Y2 is less than YAYB and having a second

[30]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[31] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2; and repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory

---

than the earliest date in 100 year-cycle in the database.

[31] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

(page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

27. Claim 68 is rejected under 35 U.S.C. as being unpatentable over Shaughnessy in view of Hazama.

As to claim 68, Shaughnessy substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly

alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[32] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the

[32]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[33] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates, by running a program on the reformatted symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a

than the earliest date in 100 year-cycle in the database.

[33] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

28. Claim 69 is rejected under 35 U.S.C. . 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 69, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[34] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the at least one date field of the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[35] of the date to the

---

[34]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[35] Shaughnessy specifically suggests that it might be desirable to set the current date to a date

corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2; sorting the reformatted symbolic representations of the dates in the form C1 C2 Y1 Y2; and running a program on the reformatted symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the

---

which compares low to all other dates (col. 7, lines 16-17 et seq).

cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

29. Claim 70 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 70, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each

of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by the symbolic representation of dates stored in the at least one date field, without the addition of any new data field to the database, and without modifying any of the symbolic representations of dates in the at least one date field, for purposes of such windowing and converting; and running a program on the converted symbolic representations of each of the dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[36] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the

[36]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

YYMMDD portion[37] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller

than the earliest date in 100 year-cycle in the database.

[37] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting of converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements the cited references by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

30. Claim 71 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 71, Shaughnessy substantially discloses the invention including the claimed 'method for representing and utilizing dates stored in at least one date field of the database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by a symbolic representation of dates stored in the at least one date field, and without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations of each of the converted symbolic representations of the dates to manipulate the dates represented by the converted symbolic representations, separately from the date data

symbolic representations contained in the at least one date field of the database' by providing a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). Shaughnessy further provides a subroutine for determining the current date[38] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy provides a subroutine that compares the YYMMDD portion[39] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy provides a subroutine for appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without modifying any of the date fields stored in the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of

---

[38]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[39] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq)

thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

Shaughnessy and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field in the database.' Booth, however, further complements the cited references by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Shaughnessy-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

31. Claims 72 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 72, Shaughnessy substantially discloses the invention including the claimed

'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'selecting a database with symbolic representations of dates stored therein according to a format wherein Ml M2 is the numerical month designator, Dl D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also a subroutine for determining the current date[40] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100

[40]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of ' determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[41] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database with the values C1 C2, Y1 Y2, M1 M2, and Dl D2 prior to collectively further processing information contained within the database associated with the respective dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory

---

than the earliest date in 100 year-cycle in the database.

[41] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

(page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

32. Claims 73 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama.

As to claim 73, Shaughnessy substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly

alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[42] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the

[42]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[43] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of the date with the values C1 C2, Y1 Y2, to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the

---

than the earliest date in 100 year-cycle in the database.

[43] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

33. Claim 74 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 74, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col. 1, lines 7-14 et seq). In particular, Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator, all of symbolic representations of dates falling within a 10-decade period of time' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines

11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses the claimed step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database' as a subroutine for determining the current date[44] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[45] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting each date in the form C1C2Y1Y2 to facilitate further processing of the dates' by appending the determined century value before the YYMMDD date in order to yield a

[44] The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

[45] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

In the event that applicant were to argue that Shaughnessy does not disclose the limitation that the two digit date is smaller or equal to the smallest date in the database, where all the dates in the database fall within a 100 year period, it would have been obvious to the ordinary skilled artisan to look to the teachings of Hazama to complement Shaughnessy's.

Hazama discloses an analogous date processing system wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to

process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq. In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

34. Claims 75 is rejected under 35 U.S.C. 103(a) as obvious over Shaughnessy in view of Hazama.

As to claim 75, Shaughnessy substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly

alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time' as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[46] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends in 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the

---

[46]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later

modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the comparison of the YYMMDD portion[47] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, MlM2, and D1 D2 in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if

than the earliest date in 100 year-cycle in the database.

[47] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

35. Claims 76 is rejected under 35 U.S.C. 103(a) as being unpatentable over Shaughnessy in view of Hazama, and further in view of Booth.

As to claim 76, Shaughnessy substantially discloses the invention including the claimed 'method of processing dates in a database' as a method and system for modifying and operating a computer system to perform operations on date fields having a two digit representation for the year without mistaking the year 2000 and the year 1900 (col.1, lines 7-14 et seq). Shaughnessy also suggests the conversion of all dates within the database from a two digit format to a four digit format as a viable, but costly alternative for the year 2000 problem (col. 1, lines 31-46 et seq). The method as recited in the body of the claim is particularly taught by the cited references as follows:

Shaughnessy discloses the step of ' providing a database with dates stored therein according to a format wherein Ml M2 is the numerical month designator, Dl D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' as a database having a 6 digit-field for storing a Date type in the form of MMDDYY(see appendix in col. 18, Date Type A), wherein the MM represents the month, DD represents the day and YY represents the year for a particular six digit-date falling within a 10-decade period of time as a date having a cycle or a range of a 100 years (col. 18, Cycle/Range C1 = THE DATE CYCLE IS 100 YEARS). As pointed out in column 2, lines 11-14 and column 3, lines 4-8 of Patent No. 5,806,063, all dates in commercial and industrial databases span within one 100 year. Shaughnessy's system being of the commercial or industrial kind described in the cited patent, must therefore, as a practical matter, inherently incorporate this limitation. Shaughnessy also discloses a subroutine for determining the current date[48] to thereby select a 100 year cycle wherein the current date is the pivot date and wherein the cycle ends a 100 years from said current date (col.5, lines 31-36). Additionally, Shaughnessy discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' as the comparison of the current date to the date when the system was installed with the modifications (modified system install date) to thereby determine the century value (col.5, lines 36-65 et seq). Alternatively, Shaughnessy discloses the

[48]The current date, by virtue of being the pivot date in the 100 year-cycle and by being initially set to the operating system date, initially set to 0000 (see col. 7, lines 19-20 et seq), is therefore not any later than the earliest date in 100 year-cycle in the database.

comparison of the YYMMDD portion[49] of the date to the corresponding date portion at the end of the 100 year cycle to thereby determine the century value (col.7, lines 7-15 et seq). Finally, Shaughnessy discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values Cl C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by appending the determined century value before the YYMMDD date in order to yield a CCYYMMDD date format, whereby said appending was performed without adding a new field to the database (col.5, lines 46-51; col.6, lines 57-65 et seq).

Shaughnessy does not particularly detail that the YaYb value for the first decade of the window is no later than the earliest Y1Y2 year designator in the database in the selection of the 10_decade window. However, Hazama discloses an analogous date processing system having two digit dates spanning from the 20th century to the 21st century stored in memory (page 2, claim 1) wherein, for a 100 year window (restricting the all the dates in the database between the 20th and 21st centuries), the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the

[49] Shaughnessy specifically suggests that it might be desirable to set the current date to a date which compares low to all other dates (col. 7, lines 16-17 et seq).

cited references. The ordinary skilled artisan having read Shaughnessy would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Shaughnessy's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Shaughnessy and Hazama do not specifically disclose the sorting the converted dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2. However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p. 939, lines 1-3 et seq). In particular, analogously to Shaughnessy, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p. 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Shaughnessy and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate Shaughnessy-Hazama's system to return the reformatted dates in chronological sequence. Such disclosed sorting would therefore be very useful when indexing the database in date order, as suggested by Booth on page 945.

36. Claims 1-3, 5, 7, 9-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over B.G. Ohms, *Computer processing of Dates Outside the Twentieth Century*, IBM Systems Journal, Volume 25, Number 2, 1986, pages 244-251, (Ohms, hereinafter), in view of Hazama.

As to claim 1, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each symbolic representation of a date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating

C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Finally, OHMS discloses the step of 'reformatting the symbolic representation of the date with the values C.sub.1 C.sub.2, Y.sub.1 Y.sub.2, M.sub.1 M.sub.2 , and D.sub.1 D.sub.2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[50], it is expressed in accordance with its corresponding century (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to

---

[50]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 2, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that 'the 10-decade window includes the decade beginning in the year 2000' by indicating that the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column).

As to claim 3, Ohms and Hazama disclose the invention as discussed in the rejection of claim 2. Additionally, Ohms discloses that the step of 'determining includes the step of determining the first value as 20 and the second value as 19' by indicating that dates that are greater or equal to the pivot date fall within the 20th century (C1C2=19) and dates that are less than the pivot date fall within the 21st century (C1C2=20) p. 2488, right hand column).

As to claim 5, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'reformatting includes the step of reformatting each symbolic representation of a date into the format C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 by indicating that dates that fall within the 20th century (greater than or equal to the pivot date) are preceded by 19 (e.g. 1925-1999), whereas

dates that fall within the 21st century (less than the pivot date) are preceded by 20 (e.g. 2000-2024) p 2477, right hand column).

As to claim 7, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses that the step of 'providing a database includes the step of converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date p 2477, table 1).

As to claim 9, Ohms and Hazama disclose the invention as discussed in the rejection of claim 1. Additionally, Ohms discloses, 'after the step of reformatting, the storing of the symbolic representation of dates and their associated information back into the database' by suggesting that converted eight dates are stored in the database although they take up more memory space than non-converted six digits dates p 2499, left hand column).

As to claim 10, Ohms and Hazama disclose the invention as discussed in the rejection of claim 9. Additionally, Ohms discloses, 'after the step of reformatting, the manipulating of information in the database having the reformatted date information therein' by suggesting that the converted dates can be saved in the database p 2499, left hand column).

37. Claims 4, 6, 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 1-3, 5, 7, 9-10 above, further in view of Booth.

As to claim 4, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 1 above. Ohms and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into p 9411, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 6, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 5 above. Ohms and Hazama do not specifically, disclose the additional

step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 8, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 1. Ohms and Hazama do not specifically, disclose the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero). However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq.) In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered

two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

38. Claims 11-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 11, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1

D.sub.2 is the numerical day designator, and Y.sub.1 Y.sub.2 is the numerical year designator, all of dates falling within a 10-decade period of time which includes the decade beginning in the year 2000' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C.sub.1 C.sub.2 for each date in the database, C.sub.1 C.sub.2 having a first value if Y.sub.1 Y.sub.2 is less than Y.sub.A Y.sub.B and having a second value if Y.sub.1 Y.sub.2 is equal to or greater than Y.sub.A Y.sub.B' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, OHMS discloses the step of 'reformatting each date in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a Y.sub.A Y.sub.B value for the first decade of the window, Y.sub.A Y.sub.B being no later than the earliest Y.sub.1 Y.sub.2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph).

Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of ' sorting the dates in the form C.sub.1 C.sub.2 Y.sub.1 Y.sub.2 M.sub.1 M.sub.2 D.sub.1 D.sub.2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight

digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 12, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Additionally, Ohms discloses step of ' converting pre-existing date information having a different format into the format wherein M.sub.1 M.sub.2 is the numerical month designator, D.sub.1 D.sub.2 is the numerical day designator and Y.sub.1 Y.sub.2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 13, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Ohms and Hazama by disclosing the claimed ' the step of selecting Y.sub.A Y.sub.B such that Y.sub.B is 0 (zero) ' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the

art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 14, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 11. Booth further complements Ohms and Hazama by disclosing the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 15, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 14. Additionally, Booth discloses the step of 'manipulating information in the database having the reformatted date therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the

Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

39. Claims 16-18, 20, 22, 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama. As to claim 16, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M1, M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-

2024) (see p 248, right-hand column. Finally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[51], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column). Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references.

---

[51]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

As to claim 17, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that 'the 10-decade window includes at least a portion of the decade beginning in the year 2000' by indicating that the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column).

As to claim 18, Ohms and Hazama disclose the invention as discussed in the rejection of claim 17. Additionally, Ohms discloses that the step of ' determining includes the step of determining the first value as 20 and the second value as 19' by indicating that dates that are greater or equal to the pivot date fall within the 20th century (C1C2=19) and dates that are less than the pivot date fall within the 21st century (C1C2=20) (p 248, right hand column).

As to claim 20, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that the step of 'reformatting includes the step of

'reformatting each symbolic representation of a date into the format C1 C2 Y1 Y2 Ml M2 Dl D2 separately from the symbolic representations in the database' by indicating that dates that fall within the 20th century (greater than or equal to the pivot date) are preceded by 19 (e.g. 1925-1999), whereas dates that fall within the 21st century (less than the pivot date) are preceded by 20 (e.g. 2000-2024) (p 247, right hand column).

As to claim 22, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses that the step of 'providing a database includes the step of converting pre-existing date information having a different format into the format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator and Y1 Y2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 24, Ohms and Hazama disclose the invention as discussed in the rejection of claim 16. Additionally, Ohms discloses, 'after the step of reformatting, the storing the symbolic representation of dates and their associated information back into the database' by suggesting that converted eight dates are stored in the database although they take up more memory space than non-converted six digits dates (p 249, left hand column).

As to claim 25, Ohms and Hazama disclose the invention as discussed in the rejection of claim 24. Additionally, Ohms discloses, 'after the step of reformatting, the manipulating of information in the database having the reformatted date information therein' by suggesting that the converted dates can be saved in the database (p 249, left hand column).

40. Claims 19, 21, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, as applied to the rejection of claims 16-18, 20, 22, 24-25 above, further in view of Booth.

As to claim 19, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16 above. Ohms and Hazama do not particularly disclose the additional step of 'sorting the symbolic representations of dates, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 21, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16 above. Ohms and Hazama do not specifically, disclose the additional

step of 'sorting the symbolic representations of dates using a numerical-order sort, after the step of reformatting.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into A corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 23, Ohms and Hazama substantially disclose the invention as discussed in the rejection of claim 16. Ohms and Hazama do not specifically, disclose the step of 'selecting includes the step of: selecting YAYB such that YB is 0 (zero).' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq.) In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for

comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting that the pivot date be set to 90 by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

41. Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 26, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall

within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2Y1Y2 M1M2, and D1 D2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however,

discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the dates in the form C1C2Y1Y2M1 M2 Dl D2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq.) In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 27, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Additionally, Ohms discloses step of ' converting pre-existing date information having a different format into the format wherein M1 M2 is the numerical month designator, D1 D2 is the numerical day designator and Y1 Y2 is the numerical year designator' by suggesting the conversion from of a date from a Gregorian format (MMDDYYYY) to a short Gregorian format (YYMMDD), wherein YY indicates the year, MM indicates the month and DD indicates the date (p 247, table 1).

As to claim 28, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Ohms and Hazama by disclosing the claimed the step of 'selecting includes the step of selecting YAYB such that YB is 0 (zero)' by suggesting that the pivot date be set to 90 and by selecting set epoch to be 1990 (i.e. YAYB = 90), such that YB equals to zero (p. 942). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of setting the pivot date to a predetermined value would enable users of the Ohms-Hazama's system to have direct

control over the return of reformatted dates to thereby preset the date processing system in accordance with their needs.

As to claim 29, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 26. Booth further complements Ohms and Hazama by disclosing, after the step of sorting, the step of 'storing the sorted dates and their associated information back into the database' by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

As to claim 30, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 29. Additionally, Booth discloses the step of ' manipulating information in the database having the reformatted dates therein' by renaming and storing the sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of the Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

42. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms, in view of Hazama.

As to claim 31, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or, greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Finally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having

the values C1 C2, Y1 Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[52], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting A window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that `the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all

---

[52]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

43. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 32, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or A 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2, is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each of the dates in the database, without the addition of any new data field to the database, with the

reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot year of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes

the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

44. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 33, Ohms substantially discloses the invention including the claimed 'method of processing symbolic representations of dates stored in a database' by presenting A computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). In particular, Ohms discloses the step of 'providing A database with symbolic

representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) would fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date would fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column). Finally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database without changing any of the symbolic representations of a date in the database during the reformatting step, with the reformatted symbolic representation of each date in the database having the values C1C2Y1Y2, in order to facilitate collectively further processing the reformatted symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century[53], it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

---

[53]Ohms implicitly discloses that C1C2 corresponds to 19 or 20 depending on whether the date is less than or greater than or equal to the pivot date.

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

45. Claims 34-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 34, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one

date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding

A new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date

with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

As to claim 35, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Additionally, Ohms discloses step of 'opening the database prior to the step of converting' by providing a subroutine to retrieve a six digit date from its storage location in an existing application program (i.e. requires opening the DB, first) prior to converting said date to an eight digit format (p 248, right hand column et seq).

As to claim 36, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation

can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 37, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 38, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 39, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string

representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq).

As to claim 40, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 41, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 42, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations according to A different data field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that A string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in A field of the database, are sorted accordingly in A different field of the database (page 839-40 et seq).

As to claim 43, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations according to a different data entry field contained in the database from the at least one date field, prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 44, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 34. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the

date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 45, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 46, Ohms, Hazama, and Booth disclose the invention as discussed in the rejection of claim 34. Ohms further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (p 248, right hand column et seq).

As to claim 47, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 35. Ohms further discloses the step of 'converting at least a substantial portion of each of the plurality of symbolic representations of dates in the at least one date field and repeating this step until each of the date data entries in the at least one date field is converted into the format that does not have the ambiguity' by converting the current date stored in the database field from an ambiguous six digit format (YYMMDD) into an unambiguous 8-digit format (CCYYMMDD), wherein the century for the date is specified (p 248, right hand column et seq).

As to claim 48, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 49, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations prior to the step of running the program on the converted symbolic representations' by suggesting that A string

representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 50, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 51, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 49. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 52, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 46. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program'

by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 53, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 47. Booth further complements Ohms and Hazama by disclosing the step of 'collectively sorting the converted symbolic representations according to a different data field in the database than the at least one date field, prior to the step of running the program' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 54, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Ohms and Hazama by disclosing the step of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 55, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Ohms and Hazama by disclosing the step

of 'collectively manipulating the converted symbolic representations' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 56, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 52. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in A field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 57, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 53. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 58, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 54. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

As to claim 59, Ohms, Hazama and Booth disclose the invention as discussed in the rejection of claim 55. Booth further complements Ohms and Hazama by disclosing the step of 'performing an operation which manipulates the data in a data field associated with the at least one date field of the database according to the converted symbolic representation of the date' by suggesting that a string representation can be used to sort and index the converted dates such that the dates appear in chronological order (page 945 et seq), whereby said dates, contained in a field of the database, are sorted accordingly in a different field of the database (page 839-40 et seq).

46. Claim 60 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 60, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of A database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; running a program on each of the converted symbolic representations of each of the respective dates to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the date data symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier

900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of A pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

47. Claim 61 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 61, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each

pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on each of the converted symbolic representations of each of the respective dates to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or A 100 year window (p 249, left-hand column; lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or

modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date

with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

48. Claim 62 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 62, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective

dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representâtions of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or A 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an

analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the

[redacted]

reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

49. Claim 63 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth..

As to claim 63, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective

dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an

analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the

reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

50. Claim 64 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 64, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective

dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program on the stored converted symbolic representations to manipulate data in the database according to the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an

analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of

the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

51. Claim 65 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 65, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of a database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date

field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248,

right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of

converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

52. Claim 66 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 66, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein M1 M2 is the numerical month

designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having A first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, M1 M2, and D1 D2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of ' selecting a window with YAYB value for A pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year

window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements

Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

53. Claim 67 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 67, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of ' providing a database with dates stored in at least one date field therein according to A format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having A first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall

within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This

determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'repeating the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted in order to facilitate collectively further processing the reformatted symbolic representations of each of the symbolic representations of each of the dates.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

54. Claim 68 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 68, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values

C1 C2, Y1 Y2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates, by running a program on the reformatted symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the at least one date field of the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the

selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

55. Claim 69 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 69, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored in at least one date field therein according to a format wherein Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the at least one date field of the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each

symbolic representation of a date in the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the at least one date field of the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the reformatted symbolic representations of the dates in the form C1 C2 Y1 Y2; and running a program on the reformatted symbolic representations of each of the dates' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

56. Claim 70 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 70, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date

field of a database utilizing symbolic representations of the dates stored in at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting; and running a program collectively on the stored converted symbolic representations to manipulate the dates represented by the converted symbolic representations, separately from the symbolic representations of dates contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the

century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and

Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

57. Claim 71 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 71, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method for representing and utilizing dates stored in at least one date field of the database utilizing symbolic representations of the dates stored in the at least one date field of the database, which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the steps of 'converting each of the symbolic representations of dates stored in the at least one date field of the database to a symbolic representation of each of the respective dates that does not create the ambiguity, by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against a pivot year, with the pivot year being less than or equal to the earliest date represented by a symbolic representation of dates stored in the at least one date field, and without the addition of any new data field to the database for purposes of such windowing and converting; and, running a program on the stored converted symbolic representations of each of the converted symbolic representations of the dates to sort or otherwise manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms

also indicates that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding A new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the selection of a pivot year for the century window, Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all

the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

Ohms and Hazama do not specifically disclose the step of 'storing the converted symbolic representations separate from the at least one date field of the database.' Booth, however, further complements Ohms and Hazama by renaming and storing sorted dates in the CUSTMER.DBF NEW_CUST.DBF databases (p. 841). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited

references. Booth's teaching of the storing the sorted reformatted dates would users of Ohms-Hazama's system to readily retrieve the reformatted dates in their chronological sequence at any time and in accordance with their needs.

58. Claim 72 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 72, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'selecting a database with symbolic representations of dates stored therein according to a format wherein Ml M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the

pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database with the values C1 C2, Y1 Y2, M1 M2, and D1 D2 prior to collectively further processing information contained within the database associated with the respective dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference,

which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

59. Claim 73 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama.

As to claim 73, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1 Y2 is the numerical year designator, all of the symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than a and having a second value if Y1 Y2 equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby

equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of the date with the values C1 C2, Y1 Y2, to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database

having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

60. Claim 74 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 74, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein Y1Y2 is the numerical year designator, all of symbolic representations of dates falling within a 10-decade period of time' by detailing a short Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of

'reformatting each date in the form C1 C2 Y1 Y2 to facilitate further processing of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a 10-decade window with a YAYB value for the first decade of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of sorting the dates in the form C1 C2 Y1 Y2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in A database to thereby derive

other dates therefrom (p 939, lines 1-3 et seq. In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

61. Claim 75 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 75, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing symbolic representations of dates stored in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1 D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short Gregorian format (MMDDYY) to

represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each symbolic representation of a date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1C2, Y1Y2, MlM2, and D1 D2 in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of 'selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the YAYB earliest Y1Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly

detail that the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. The ordinary skilled artisan having read Ohms would immediately see the need to determine which 100 year span to use. This determination would have led the ordinary skilled artisan to the Hazama reference, which teaches the pivot date being smaller than the smallest two digit date in the database having all the dates within a 100 year period as a solution to restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

62. Claim 76 is rejected under 35 U.S.C. 103(a) as being unpatentable over Ohms in view of Hazama, further in view of Booth.

As to claim 76, Ohms substantially discloses the claimed invention. In particular, Ohms discloses the claimed 'method of processing dates in a database' by presenting a computer-implemented method for processing date outside the twentieth century (see title, p 244 et seq). Ohms further discloses the step of 'providing a database with dates stored therein according to a format wherein Ml M2 is the numerical month designator, Dl D2 is the numerical day designator, and Y1 Y2 is the numerical year designator' by detailing a short

Gregorian format (MMDDYY) to represent dates (p 247, see table 1), wherein said dates fall within a 10 decade or a 100 year window (p 249, left-hand column, lines 3-7 et seq) and wherein the 100 year window contains dates that span in the 21st century (2000-2024) (p 248, right hand column). Ohms also discloses the step of 'determining a century designator C1 C2 for each date in the database, C1 C2 having a first value if Y1 Y2 is less than YAYB and having a second value if Y1 Y2 is equal to or greater than YAYB' by indicating that years that are later or equal (25-99) to the pivot date (25) fall within the 20th century thereby equating C1C2 to 19 (i.e. 1925-1999), whereas dates that are earlier 900-24) than the pivot date fall within the 21st century thereby equating C1C2 to 20 (i.e. 2000-2024) (see p 248, right-hand column. Additionally, Ohms discloses the step of 'reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C1 C2, Y1 Y2, Ml M2, and Dl D2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates' by indicating that upon determining that a two-digit date falls within the 20th or the 21st the century, it is expressed in accordance with its corresponding century without actually adding a new data field or modifying the database (i.e. 25-99 -----> 1925-1999, and 00-24 -----> 2000-2024) (p 248, right hand column).

Regarding the step of ' selecting a window with a YAYB value for a pivot date of the window, YAYB being no later than the earliest Y1 Y2 year designator in the database,' Ohms discloses specifying a year as the desired starting point (pivot date) of the 100 year window (p 248, right hand column, 2nd paragraph). Ohms does not particularly detail that

the pivot date is earlier than the earliest two digit date in the database. Hazama, however, discloses an analogous date processing system wherein, for a 100 year window, the pivot date for the window is selected based on a two digit year date that is smaller than the smallest two digit year date in the database (e.g. if smallest two digit date stored in database is 73, the pivot date for the 100 year window is chosen to be 72). (page 4 of translated document, last paragraph). It would have been obvious to one of ordinary skill in the art of data processing to combine the teachings of the cited references. Hazama's teaching of the pivot date being smaller than the smallest two digit date in the database would restrict the selection of Ohms's window and thereby forcing all dates already stored in the database to fall in the 20th century.

Further, Ohms and Hazama do not specifically, disclose the step of 'sorting the dates in the form C1 C2 Y1Y2 M1M2 D1D2.' However, Booth discloses an analogous system that utilizes the Clipper programming language to process dates stored in a database to thereby derive other dates therefrom (p 939, lines 1-3 et seq.) In particular, analogously to Ohms and Hazama, Booth discloses the SET EPOCH command for comparing an entered two digit date with the year digit (pivot date) of the epoch setting to determine the century to place the date into (p 941, see SET EPOCH paragraph et seq) thereby converting a six digit date (MMDDYY) into a corresponding eight digit date (C1C2Y1Y2M1M2D1D2) (see p. 940-941). Additionally, Booth complements Ohms and Hazama by suggesting the sorting of converted dates after having been reformatted by the SET EPOCH command (p. 945). It would have been obvious to one of ordinary skill in the art of data processing at the time of the instant invention to combine the teachings of the cited references. Booth's teaching of

sorted reformatted dates would facilitate the Ohms-Hazama's system to return the reformatted dates in chronological sequence. And, it would therefore be very useful when indexing the database in date order, as suggested by Booth in page 945.

REMARKS

Applicant's arguments will be addressed in the same order in which they are presented in the remarks.

1. Applicant argues that Shaughnessy does not teach or suggest " *the step of selecting a 10-decade window YaYb no later than the earliest Y1Y2 year designator in the database.*" Applicant alleges that Shaughnessy only discloses the selection of a 10 decade window utilizing the date the system was installed. In response, the Examiner respectfully submits that Shaughnessy teaches the selection of a 10-decade window in figure 4 and the necessity of such a window starting with a date no later than the earliest year in the database is taught in Hazama.

2. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests "*the step of determining a century designator C1C2 for each symbolic representation of a date in the database, C1C2 having....*" Applicant alleges that the teaching of Shaughnessy or Hazama is to determine a century designator for at most two date representations being processed in a called subroutine at a given time. In response to the preceding argument, the Examiner respectfully submits that even under the allegation above, the Shaughnessy-Hazama

combination would still disclose the claimed limitation as long as the references teach or suggest the determination of a century designator for each date in the database. As discussed in the office action, Shaughnessy determines a century designator for converting a current date from a six-digit to an eight digit format before the converted date is returned for use in a particular application. Shaughnessy determines the century value (19 or 20) by comparing the current date to the corresponding date portion when the system was installed with the modifications. Further, Shaughnessy suggests that the above approach can be used to determine a century designator for converting each six digit date in a database to corresponding eight digit dates. However, Shaughnessy refrains from such an approach, though capable of curing the year 2000 problem, on economic instead of technical grounds, since it might not be cost efficient. To the extent applicant is arguing that Shaughnessy fails to extrapolate the operation of date conversion from a single instance to an entire database, it is first noticed that one of ordinary skill in the art extrapolates single operations to batch processing of an entire database as a matter of automation efficiency, it is secondly pointed out that Shaughnessy teaches that its date conversion processing would be inserted for every occurrence of date processing, i.e. across the entire input gamut, col. 4 lines 27 to 33, and it thirdly noticed that Shaughnessy even provides a specific example of checking the due dates in a database for being overdue col. 4 lines 38 to 43. Further, Hazama complements Shaughnessy by disclosing the use of a pivot date that is smaller than any other date in the database to compare each date in the database with the pivot date to thereby determine whether each two digit year in the database should be preceded by 19 or 20. Therefore, the

Shaughnessy -Hazama combination does teach the above limitation, as claimed.

3. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests the step of "*reformatting the symbolic representation of the date with the values C1C2,Y1Y2,M1M2, and D1D2 to facilitate further processing of the dates.*" Applicant alleges that the teaching of Shaughnessy or Hazama is to reformat two dates at a time in the called result of the processing of the two reformatted date data entries, and not to facilitate further processing of the dates by reformatting the symbolic representations of the dates (claim 4). In response to the preceding argument, the examiner respectfully submits that the Shaughnessy- Hazama combination does disclose the reformatting of the dates in the C1C2Y1Y2M1M2D1D2 format to facilitate the further processing of these dates. Shaughnessy's conversion of the current date of an operating system from a six digit format to an eight digit format each time said date is going to be used in application. Such reformatted dates are further utilized by returning one date field with the converted date to the subroutine and by returning a parameter to the application program for use in further operations. As explained above, Shaughnessy suggests that such approach can be extended to reformat dates already stored in database such that they can be used for further processing. Therefore, the Shaughnessy-Hazama combination does teach the above limitation, as claimed.

4. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests the step of *"sorting the symbolic representations of the dates."* Applicant alleges that Shaughnessy or

Hazama only teaches the comparison of two dates to each other in the called subroutine and returning to the program an indication of the result of the comparison. In response to the preceding argument, the examiner respectfully submits that it was conceded that Shaughnessy and Hazama do not teach the step of sorting the symbolic representations of the dates. However, the Examiner relied upon the Booth reference for such teaching, as detailed in paragraph 10 et seq of the office action. It is noted that Applicant fails to address and rebut the rejection of claim 4 over Shaughnessy, Hazama and Booth. Therefore, the issue is considered to be waived and the rejection of claim 4 is sustained. **See In re Berger**, Slip Op 01-1129.

5. Applicant argues that neither Shaughnessy or Hazama teaches or suggests the step of *"reformatting each symbolic representation of a date in a format C1C2Y1Y2M1M2D1D2 (claim 5), nor sorting the symbolic representations of dates in numerical order sort (claim 6), nor storing the symbolic representation of dates and their associated information back into the database (claim 9); nor manipulating information in the database having reformatted date information therein (claim 10)."* In response to the preceding argument, the examiner respectfully submits that with regards to claim 5, Shaughnessy discloses the limitations as discussed above in paragraph 3 of the remarks. Regarding claim 6, Shaughnessy, Hazama and Booth disclose the cited limitation, see discussion above in paragraph 4 of remarks. Regarding the limitation of claim 9, Shaughnessy discloses the step of storing the symbolic representation of dates and their associated information back into the database, as discussed

in the office action. Shaughnessy teaches the storing in the database of current date after it has been converted from the six digit format to the eight digit format. Further, Shaughnessy suggests that such an approach can be extended to dates in a database. Consequently, Shaughnessy discloses the claimed limitation of claim 9. Regarding claim 10, Shaughnessy and Hazama disclose the cited limitations as discussed above in paragraphs 3 and 4 of the remarks.

6. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests the step of "converting pre-existing date information [within a database] having a different format into the format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and Y1Y2 is the numerical year designator (claim 7). In response to the preceding argument, the Examiner respectfully submits that Shaughnessy does disclose the cited limitation. In particular, Shaughnessy discloses the conversion of a current date from a six digit format (YYMMDD) to an eight digit format (CCYYMMDD). Shaughnessy also suggests that the preceding approach could be extended to convert dates already stored in a database.

7. Applicant argues that neither Shaughnessy nor Hazama teaches or suggests the step of selecting YaYb such that Yb is 0 (zero) (claim 8). In response to the preceding argument, the examiner respectfully submits that it was conceded that Shaughnessy and Hazama do not teach the step of selecting YaYb such that Yb is zero. However, the Examiner relied upon

the Booth reference for such teaching, as detailed in paragraph xxx of the office action. It is noted that Applicant fails to address and rebut the rejection of claim 8 over Shaughnessy, Hazama and Booth. Therefore, the issue is considered to be waived and the rejection of claim 8 is sustained. **See In re Berger**, Slip Op 01-1129.

8. Applicant argues that Booth does not teach or suggest the step of *"selecting a 10-decade window YaYb no later than the earliest Y1Y2 year designator in the database."* Applicant alleges that Booth merely selects, e.g. "nyear" in order to "handle dates that use only two digits for the year [w]hen a two-digit year is entered into a date [by comparing] its year digits... with the year digits of the epoch setting to determine the century...(Id. at 941). Therefore, Applicant contends that Booth does not disclose the earliest Y1Y2 year designator in the database. In response to the preceding arguments, the Examiner respectfully submits that Booth was not relied upon for the teachings of the cited limitations. Rather Hazama was relied upon for such teaching. Therefore, Applicant's argument is not relevant.

9. Applicant argues that Booth does not teach or suggest the step of *"the step of determining a century designator C1C2 for each symbolic representation of a date in the database having...."* Applicant alleges that:

"there is no need to determine a century designator for each symbolic representation of a date in Booth's database since each is already stored with the century designator included in the date datum so stored in integer format. In addition, the teaching of Booth is to determine a century designator on an individual date datum basis for date data entry,

date display, incrementally determining a date based upon a given initial date datum , etc. This calling of certain functions disclosed by Booth to, for example, display a date, or compare two dates, or increment a date from a starting date, are virtually identical to the pertinent disclosure in Shaughnessy."

In response to the preceding arguments, the Examiner respectfully submits that Booth was not relied upon for the teachings of the cited limitations. Rather Shaughnessy was relied upon for such teaching. Therefore, Applicant's argument is not relevant.

10. Applicant argues that Booth does not teach or suggest the step of *"reformatting the symbolic representations of the date with the values C1C2Y1Y2M1M2 and D1D2 to facilitate further processing of the date.*" Applicant alleges that:

"Booth, like Ohms, does not need to do the recited reformatting, since the dates stored in the database in their original format already contain all the information needed to determine the four digit designation of the date, including the century of the particular date datum. The process of the claimed invention is not needed for dates stored with the century designator already known from what is stored and the Y2k ambiguity not present. Furthermore, the teaching of Booth, like Shaughnessy, is to reformat one or two dates at a time in a called Clipper date functionality and the return to the program from the called subroutine with information resulting from the performance of the programming functionality, e.g., an input to a display, a result of a comparison, a newly calculated date, etc. Booth does not teach facilitating "further processing of the dates" by "reformatting the symbolic representation of the date" "for each symbolic representation of a date in the database."

In response to the preceding arguments, the Examiner respectfully submits that Booth was not relied upon for the teachings of the cited limitations. Rather Shaughnessy was relied upon for such teaching. Therefore, Applicant's arguments are not relevant.

11. Applicant argues that Booth does not teach or suggest the step of *"sorting the symbolic representations of dates"*, as recited in claim 4. Applicant contends that:

> "Whatever sorting Booth teaches does not need to first reformat the date data, since the integer format can be and is sorted in its initial format. The method of the claimed invention, including the reformatting steps is simply not relevant to a database that stores date data as Clipper does, in integer format, as described in Booth."

In response to the preceding arguments, the Examiner respectfully submits that Applicant's reading of Booth is inaccurate. Booth discloses the sorting and indexing of dates stored in a database after said dates have been converted to an eight digit format (YYYYMMDD). See pages 845, 945. Therefore, the rejection of claim 4 is proper.

12. Applicant argues that Booth does not teach or suggest the step of *"reformatting each symbolic representation of a date into a format C1C2Y1Y2M1M2D1D2"* as recited in claim 5. Applicant further argues that Booth does not disclose the step of *"sorting the symbolic representations of dates using a numerical order sort"* as recited in claim 6. Additionally, Applicant argues that Booth does not disclose the step of *"storing the symbolic representation of dates and their associated information back into the database"* as recited in claim 9 nor the step of *"manipulating information in the database having the reformatted date information therein"*, as recited in claim 10. In response to the preceding arguments, the Examiner respectfully submits that Booth was not relied upon for the teachings of the limitations in claims 5, 9 and 10. As discussed above in the remarks, Shaughnessy was relied

upon for such teachings. Further, regarding claim 6, it was previously discussed above that Booth teaches such limitation at pages 845 and 845. Therefore, the rejection of claims 5, 6, 9 and 10 is proper.

13. Applicant argues that Booth does not teach or suggest the step of "*converting pre-existing date information having a different format into the format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and Y1Y2 is the numerical year designator,*" as recited in claim 7. Applicant alleges that the fact that Booth teaches converting date format into the "recited format" does not teach it as part of the process of the claimed invention. In response to the preceding arguments, the Examiner respectfully submits that Booth was not relied upon for such teaching. Rather, Shaughnessy was relied upon the limitations of claim 7. Consequently, Applicant's arguments are not relevant.

14. Applicant argues that Booth does not teach or suggest the step of "*selecting YaYb such that Yb is 0 (zero),*" as recited in claim 8. Applicant alleges that even though SET EPOCH can and does use pivot years ending in 0, it is not a process according to the claimed invention. In response to the preceding arguments, the Examiner respectfully submits that Booth does disclose Yb to be zero by selecting YaYb to be equal to 90. See page 942. It is noted that Applicant's arguments that Booth's teaching is not a process according to the claimed inventions fails to comply with 37 CFR 1.111(b) because they amount to a general

allegation that the claim define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Applicant simply alleges that the cited limitations are not taught by Shaughnessy without actually explaining how these limitations are distinguishable from the corresponding portions in Booth on which the Examiner relied to establish the prima facie case. See page .....of the office action. Consequently, Applicant has failed to successfully rebut the rejection of claim 8. ....Generally, Applicant bears the burden of explaining why the evidence on which the Examiner relies is insufficient to establish a prima facie case or demonstrating that Applicant has provided evidence which rebuts the prima facie case. See In re Rouffet, 149 F.3d 1350, 1355 47 USPQ2d 1453, 1455 (Fed. Cir. 1998). Furthermore, Shaughnessy's process would select a Yb value of 0 for one year out of every 10 when operated with daily update, col. 6 lines 4 to 45.

15. Applicant argues that Ohms does not teach or suggest the step of *"Providing a database with symbolic representations of dates stored therein according to a format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and Y1Y2 is the numerical year designator, all of the symbolic representations falling within a 10-decade period of time, as recited in claim 1.* Applicant alleges that Ohms does not disclose the above limitations since Ohms teaches providing a database with the dates in a Lilian format. In response to the preceding arguments, the Examiner respectfully submits that Applicant's reading of Ohms is incorrect. Ohms teachings are not limited to dates in

Lilian format. As discussed in the office action, Ohms discloses the storing of dates in a database in Gregorian format, wherein said dates are converted from a six digit format (YYMMDD) to an eight digit format(YYYYMMDD). See page 247, table 1. Ohms further teaches that the dates stored in the database do fall within a ten decade period. See page 249. Consequently, the rejection is proper.

16. Applicant argues that Ohms does not teach or suggest the step of "selecting a YaYb value for the first decade of the window, YaYb being no later than the earliest Y1Y2 year designator in the database. Applicant contends that, at best, Ohms teaches or suggests a YaYb based upon dates that are currently being input into the database. In response to the preceding arguments, the Examiner respectfully submits that it was conceded in the office action that Ohms does not detail the cited limitation. However, Hazama was relied upon to complement Ohms for its teaching of a pivot year date that is smaller than the smallest two digit year date in the database. Therefore, the claimed limitation is taught by the Ohms-Hazama combination. Consequently, the rejection is proper.

17. Applicant argues that Ohms does not teach or suggest the step of "determining a century designator C1C2 for each symbolic representation of a date in the database, C1C2 having. Applicant contends that Ohm teaches entering date data into the database to be converted into Lilian format for storage and manipulation within the database. Applicant further alleges that since the conversion in Lilian format does not require the determination

of a century designator for data in the database, then Ohms cannot teach such limitation. In response to the preceding arguments, the Examiner respectfully submits that Applicant misread Ohms' teachings. As pointed out above, Ohms' teachings are not limited to conversion in Lilian format. Ohms also discloses the conversion of dates stored in a database in Gregorian format from a six digit format to an eight digit format to include the century designator. See page 247, table 1 and page 248.

18. Applicant argues that Ohms does not teach or suggest the step of "reformatting the symbolic representation of the date with the values C1C2, Y1Y2, M1M2, and D1D2 to facilitate the further processing of the dates. Applicant contends that Ohms does not disclose such limitation since it teaches reformatting into Lilian format and thereafter processing the date data in the database utilizing the Lilian format. In response to the preceding arguments, the Examiner respectfully submits that, as pointed out above in the remarks, Ohms's teachings are not limited to reformatting in Lilian format. Ohms discloses the reformatting of a short Gregorian date having six digit into a Gregorian date having eight digits. See page 247, table 1.

19. Applicant argues that Ohms does not teach or suggest the steps of sorting the symbolic representations of dates (claim 4); or reformatting each symbolic representation of a date into the format C1C2Y1Y2M1M2D1D2 (claim 5) or sorting the symbolic representations of dates and their associated (claim 6) or storing the symbolic representation

of dates and their associated information back into the database (claim 9) or after the step of reformatting, manipulating information in the database having the reformatted date information therein (claim 10) or converting pre-existing date information having a different format into the format wherein M1M2 is the numerical month designator, D1D2 is the numerical day designator and Y1Y2 is the numerical year designator or selecting YaYb such that Yb is 0 (zero) (claim 8). In response to the preceding arguments, the examiner respectfully submits that it was conceded in the office action that Ohms does not teach the limitations of claims 4, 6, and 8. However, Booth was relied upon to complement Ohms' teachings in order to reject the cited claims. Regarding claims 5, 9 and 10, it was pointed out in the office action that Ohms teaches the reformatting of short order Gregorian dates having six digits into Gregorian dates having eight digits to thereby store the converted dates in the database for further use and processing. The limitations of these claims were fully addressed in the office action. It is noted, however, that Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the references. Applicant simply alleges that the cited limitations are not taught by Ohms without actually explaining how these limitations are distinguishable from the corresponding portions in Ohms on which the Examiner relied to establish the prima facie case. See page .....of the office action. Consequently, Applicant has failed to successfully rebut the rejection of claims 4-10 as laid out in paragraph ..Generally, Applicant bears the burden of explaining why the evidence on which the Examiner relies is insufficient

to establish a prima facie case or demonstrating that Applicant has provided evidence which rebuts the prima facie case. See In re Rouffet, 149 F.3d 1350, 1355 47 USPQ2d 1453, 1455 (Fed. Cir. 1998).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner

should be directed to Jean R. Homere whose telephone number is (703)-308-6647. The examiner can normally be reached on Monday-Friday from 09:30 a.m.-6:00 p.m. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Breene, can be reached on Monday-Friday from 8:00 a.m. to 3:30 p.m. at (703)-305-9790.

Any response to this action should be mailed to: Commissioner of Patents and Trademarks Washington, D.C. 20231, **or faxed to:** (703) 746-7239, (for formal communications intended for entry), **or faxed to:** (703) 746-7238, (for after final communications intended for entry), **Or:** (703) 746-7240(for informal or draft communications, please label "PROPOSED" or "DRAFT"). Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA., Sixth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-3900.

Jean R Homere

Jean R. Homere
Primary Examiner, A.U. 2177
June 20, 2002

JEAN R. HOMERE
PRIMARY EXAMINER